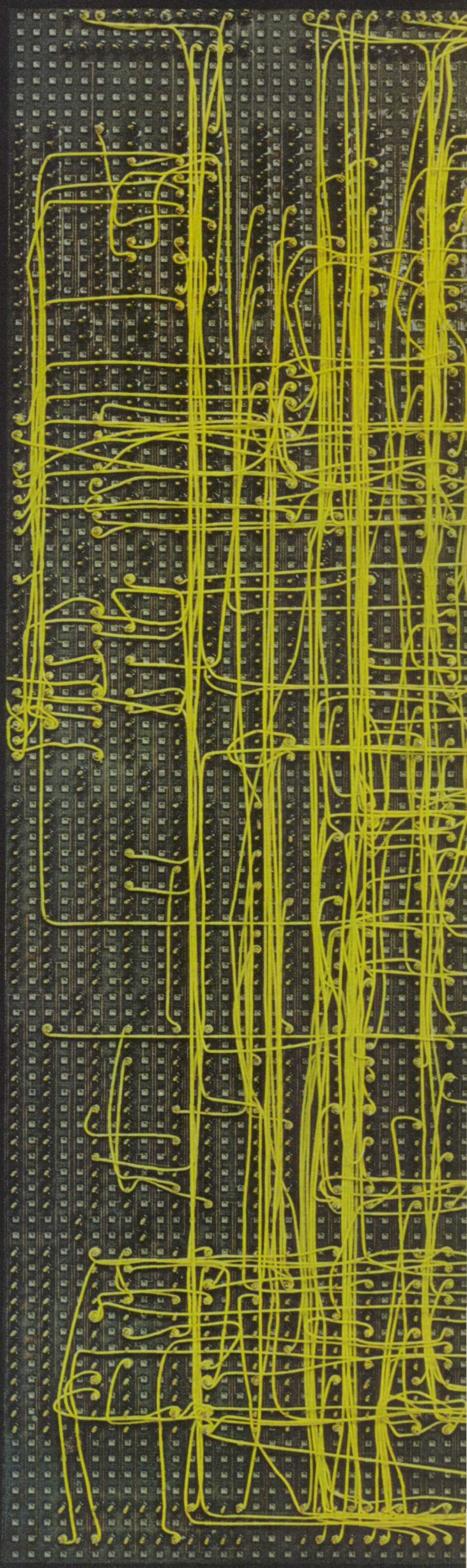
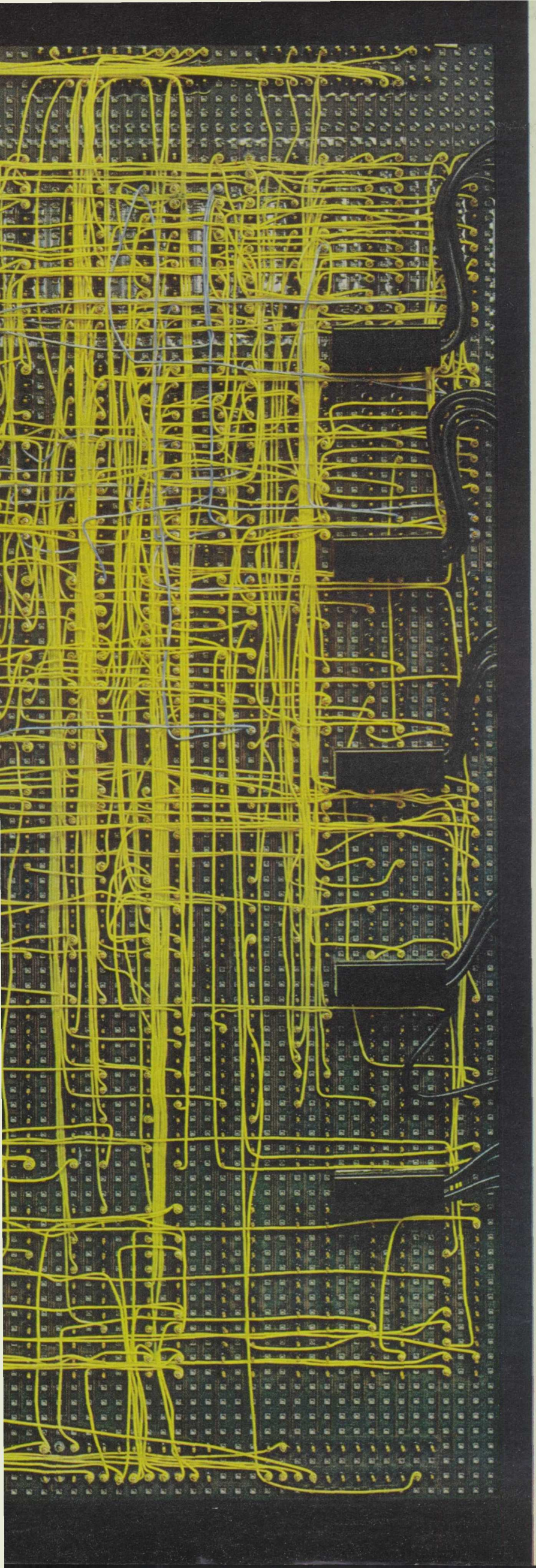
$$\begin{array}{r} 2 \\ +\,1 \\ \hline 3 \end{array}$$

How the Computer gets the answer



The delicate array of wires and pins on a panel taken from one of IBM's newest machines suggests the enormous intricacy and precision of a modern computer's inner circuitry.



Step by step, an easy exercise reveals the workings of man's most complex machine

Two plus One—not exactly a problem to set the mind racing or to blow a computer's fuse. Yet it is enough to send electric pulses flying through the computer's intricate web of wires. Although we are barely in the third decade of the computer age, computers already touch the life of everyone in the U.S. Each year—each day—our involvement with these machines rises toward unimaginable levels. It is a commonplace that if it weren't for computers we couldn't fly to the moon, or even keep an accurate record of the national debt. On the question of *how* it does what it does, however, the computer has always remained essentially mysterious—unfathomable to all but a small handful of initiates. An officer of one major computer concern guessed recently that not more than 2% of his employes really know how it works.

On the following pages LIFE presents a computer primer which, with the help of models, shows clearly what goes on inside a computer as it tackles one simple problem in addition. In just the same distinctively straightforward fashion, the computer would approach any problem, no matter how complex. On the pages following the problem, a new-style computer is opened up for inspection, with a close-in view of some of its sophisticated and surprisingly beautiful components.

Despite all recent improvements, a computer still cannot "think." It can, in fact, do only three rather simple things, but it can do them superlatively well: 1) it can retrieve almost instantly any stored piece of information—the cost of the item you charged at the department store this morning, a number needed to calculate a rocket's trajectory; 2) it can compare two numbers and perform any mathematical operation with them—add, subtract, multiply or divide; 3) it can perform any combination of these functions in a specified sequence—a program—without additional human intervention. The paradoxical essence of a computer is its simplicity: it takes but a single step at a time. When it completes one tiny operation, it goes on to the next and then to the next, all at bewildering speed, until finally it gets the answer—in this case 2+1=3.

Photographed by
HENRY GROSKINSKY

Text by
ROBERT CAMPBELL

CONTINUED

'On-off' language feeds in

The modern digital computer, whether it is used for routine bookkeeping or for complex scientific calculations, is composed of half a dozen basic elements. The same elements form the components of the ingenious photo models shown here and on the following pages. Like its big electronic brother, the model is constructed to handle a variety of calculations. In the case at hand it will tackle the problem of adding 1 and 2. As the model proceeds toward the answer, it will reveal how a big digital machine moves step by step toward its own solutions.

The first requirement of any computer is an Input—a means of feeding into the machine the information, data and instructions it will need to solve a problem. This can be done in a variety of ways, including magnetic tape, an electric typewriter hooked into the computer, or the familiar punched "do-not-fold-mutilate-or-spindle" card shown in the photo model, on which information is contained in coded holes punched in the card. The Input must be translated into language the machine can understand. This process and the language involved will be explained further on, but the step is symbolized here by the glowing disks labeled "Encoder."

Information that is put into the computer must be stored someplace until the machine is ready to make use of it. This is the function of a computer's Memory section—represented here by the vertical boxes.

Then, when the computer begins to work on the problem, an Arithmetic unit—which actually does the calculation—comes into play. So also does a Control system to govern the machine's sequence of operations. (Both these elements appear on the next page.) After the problem has been solved and the answer stored (suggested here by the Answer box), the computer must then reverse its opening procedure. It must retranslate its machine language and display the answer in a form the person who presented the problem can understand.

The first problem, obviously, is language. How does an electronic instrument handle and manipulate numbers?

The answer is suggested by the nature of all electrical devices: a light bulb is either on or off, a switch is either open or closed, a magnet has a field in one direction or the opposite. For the purpose of understanding computer language, one can think of the "on" condition as being equal to 1 and the "off" state as 0. So the computer, which is made up of literally millions of electronic components, has two numbers it can work with. These numbers, 1 and 0, form all the elements needed in the binary system of notation.

In our more familiar decimal system, the right-hand column of a figure counts numbers up to 10; the column to the left of that registers the number of 10s; the column next to the left registers hundreds—then thousands, millions and so on. In binary notation, the columns starting at the right register powers of 2 instead of 10. Take the binary number 10110, with the successive powers of 2 noted above each column:

16	8	4	2	1
1	0	1	1	0

Adding together the powers of 2 turned "on" in this binary number —16, 4 and 2—we arrive at its decimal equivalent—22. The first eight decimal numbers, translated into the binary system look like this:

1= 1	5= 101
2= 10	6= 110
3= 11	7= 111
4=100	8=1000

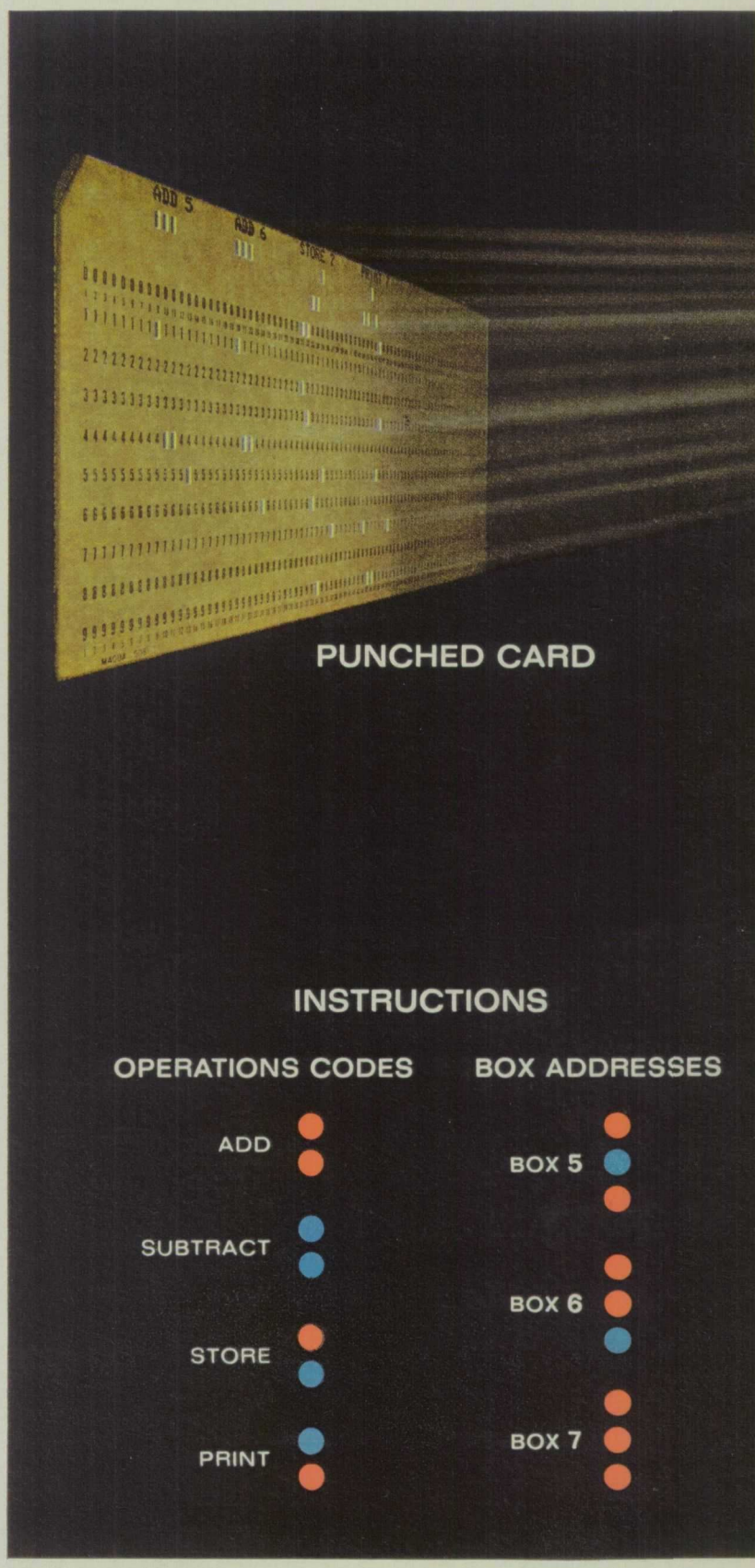
In a real computer (which would encounter the decimal number 6 as a series of pulses —"on-on-off") binary coded information is stored electronically in memory units of enormous sophistication.

Instead of electric pulses, however, our model uses colored disks, with red disks representing 1 (or "on") and the blues 0 (or "off"). Our Memory unit is a series of simple boxes—but the principle is exactly the same. The boxes in the model have room for "words" of exactly five binary digits—or "bits"—but this is, in fact, large enough for the model to add together any two numbers whose total is not larger than 31.

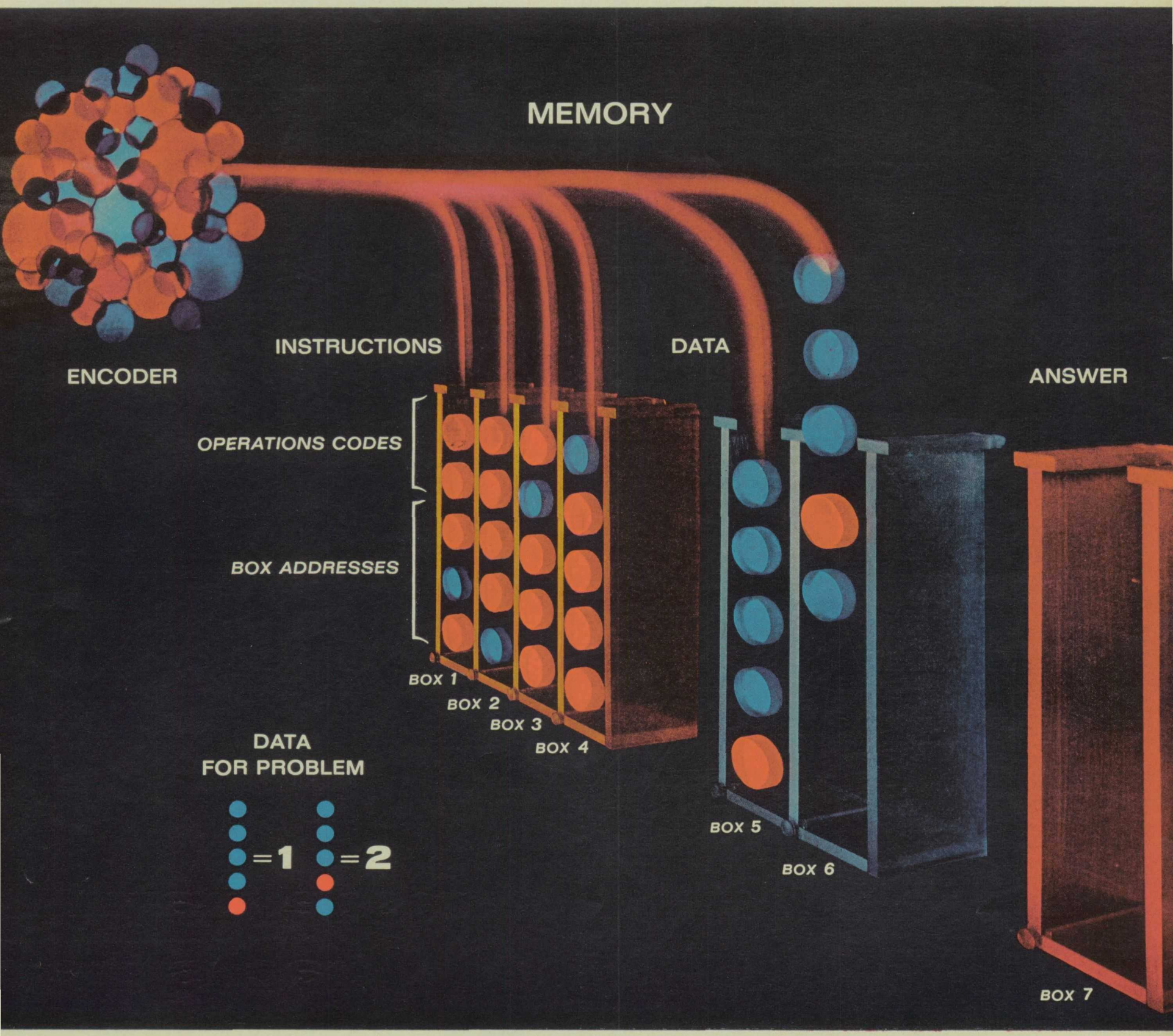
The binary number system can provide more than pure numerical data for the machine. Depending on an agreed-upon code, the combinations of bits can stand for letters of the alphabet, symbols, punctuation marks and so forth. Instructions to the machine are also given in abbreviated binary form. A relatively brief "word" can tell the computer to take two numbers stored in its memory, add them together, subtract a third number from the total and store the result.

The photo model follows this arrangement exactly as it proceeds with the task of loading in the data and instructions it will need to solve the problem of adding 1 and 2. The data, in this case the two numbers we wish to add together, goes in first. The Encoder translates the two numbers into binary notation, using red disks for 1 and blue disks for 0. Thus the number 1 is stored in Box 5 of Memory as a five-bit "word" consisting of a red disk followed by four blue ones. Into Box 6 goes the number 2—a blue disk, a red one and then three blue disks as indicated by the data key. Each piece of information that enters a computer is allocated a specific place in the Memory bank, so that the human operator of the machine can later direct the computer to go back and get it.

With the data stored away in the model's Memory, the machine needs instructions as to what should be done with it. These instructions are entered in the Memory boxes numbered 1 to 4. Each five-bit instruction consists of a



the data and instructions



three-bit Box Address number telling where to find the needed data, and a two-bit Operations Code which specifies what should happen to the data. In the Address section of Box 1 are a red, blue and red disk, the model's binary code for the number 5. In the Operations section are two red disks that, as indicated by the key, mean "add." In the language of professional computer people, Box 1 now contains the cryptic directive: "Add 5." What the instruction really means is: "Take the contents of Box 5 and carry it to the Arithmetic unit for addition." Box 2 holds a similar instruction regarding Box 6. Boxes 3 and 4 contain instructions on where the computer should store the answer once it has been obtained, and how the answer should be displayed. These two operations will be explained on the next page.

The instructions as to how to proceed are long and very detailed, for this model and all real computers are excessively simple-minded. They must be guided every step of the way. But they are also indefatigable and will follow the instructions endlessly, until the problem is solved.

Like the family dishwasher, our model is now loaded with everything it needs to cope with the problem—both data and the instructions it needs. When the button is pushed no further human intervention will be necessary.

CONTINUED

The model train on these pages shows how a real computer carries out its task. The tracks, turntables and cranes of the railroad system are remarkably similar in function to the wiring and elaborate electronic switching circuits of a modern digital computer. Furthermore, the train will accomplish the task of adding 1 and 2 in almost precisely the same way that a computer would add any two numbers: taking steps one at a time in an ordered sequence.

The boxes containing the computer's Instructions, Data and, eventually, the Answer are here separated to show the actual process more clearly. The remaining major elements of the computer have been added: 1) the Control section at center makes certain that instructions are carried out in proper sequence; 2) the Arithmetic unit at lower right will ultimately combine the two numbers in our problem.

Starting at the Instructions area, the train picks up its first load, consisting of five colored disks from Box 1. These disks constitute an order: "Take the contents of Box 5 and deliver it to the Arithmetic unit for addition." The three disks—red, blue, red—already on the train make up the Box Address for Box 5. The two red disks just about to land on the train make up the Operations Code for addition.

Thus loaded, the train proceeds toward the Control section. At the lower turntable the two red disks of the operational order are unloaded and the order "add" is read and remembered. The train moves to the upper turntable, where the Box Address is taken off. Guided by this address, the upper turntable sends the train out on Track 5 to its destination, Box 5. Here it picks up the box's contents, which are part of the data for the problem, in this case the number 1.

With this cargo, the train circles back to the Operations Codes turntable where, remembering the "add" order already deposited there, the turntable sends it out on the track labeled "Add" to the Arithmetic unit. There five disks signifying the number 1 are lifted off and put in the Accumulator, the component of the Arithmetic unit that performs arithmetical operations. The disks arrived on the Add track, so they will be added.

Since the instruction has been carried out, the train now loops up and around and returns to the Instructions area. There it receives the orders for its second trip from Box 2. These instructions tell it to take the contents of Box 6 to the Accumulator for addition. As on its first trip, the train goes to Control where the two disks meaning "add" are first unloaded at Operations; then at the Box Addresses turntable the Box Address disks are taken off, and the turntable sends the train off on Track 6 to Box 6. Here it picks up the number 2— a blue, a red and three blue disks—and, as before, proceeds to the Operations turntable, where the instruction "add" is in force. The turntable sends the train down the Add track again, and the number 2 is loaded into the Accumulator where the number 1 was previously stored.

The Accumulator, which keeps a running tally of whatever is put into it, instantly adds the new number to the old and produces a total. In a real computer, this trick is accomplished with logic circuits. Because they are electronic and thus always either "on" (1) or "off" (0), the circuits can follow absolutely the simple rules of binary addition. Incoming pulses are combined, through the logic circuits, to produce a new condition according to this scheme: $0+0=0$, $0+1=1$, $1+0=1$, $1+1=10$.

Thus, in our problem:

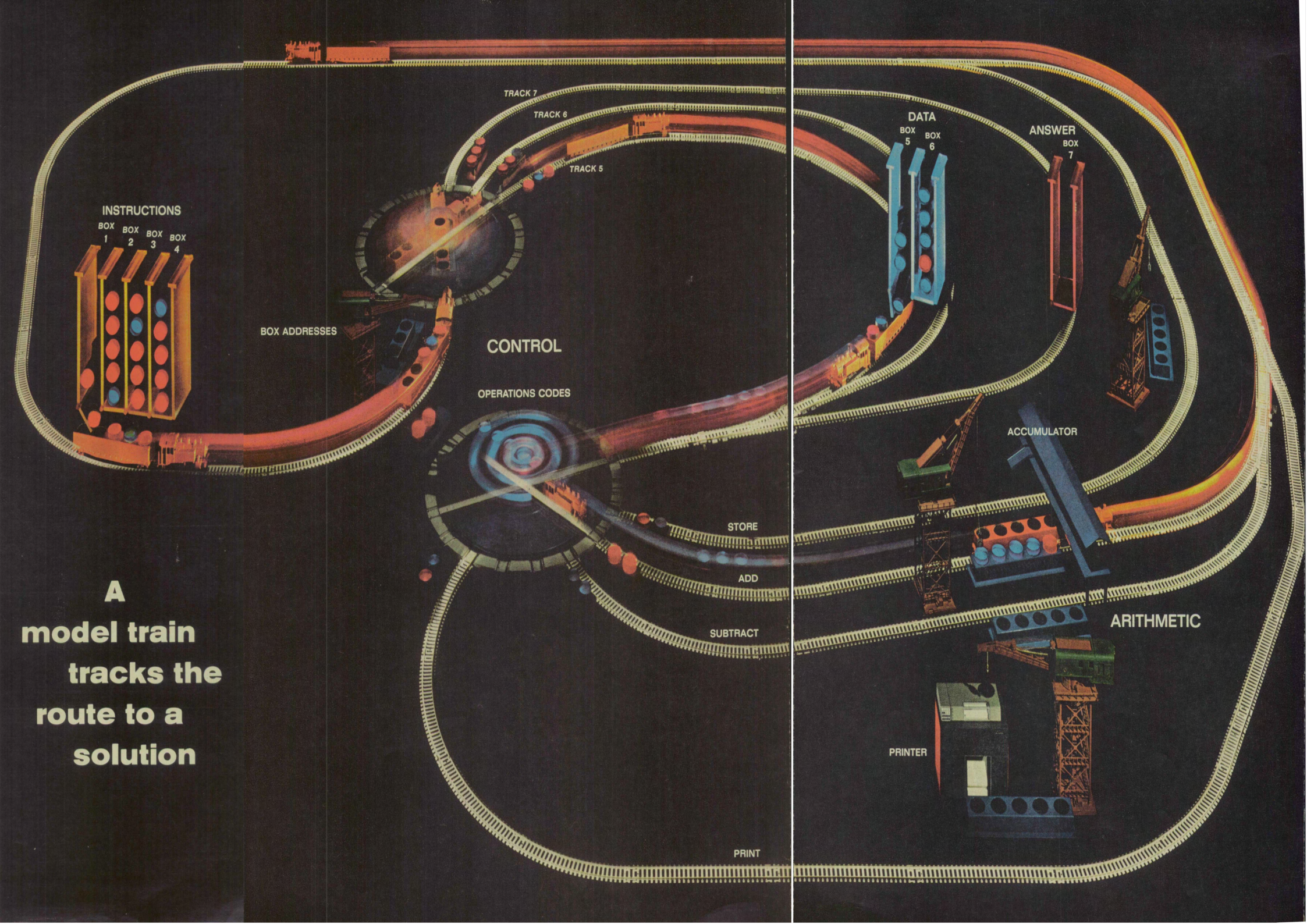
$$\begin{array}{r} 00001=1 \\ \underline{00010=2} \\ 00011=3 \end{array}$$

The logic circuits can carry numbers along from one column to the next, just as in decimal addition. This would have been necessary if our problem had been

$$\begin{array}{r} 00011=3 \\ \underline{00001=1} \\ 00100=4 \end{array}$$

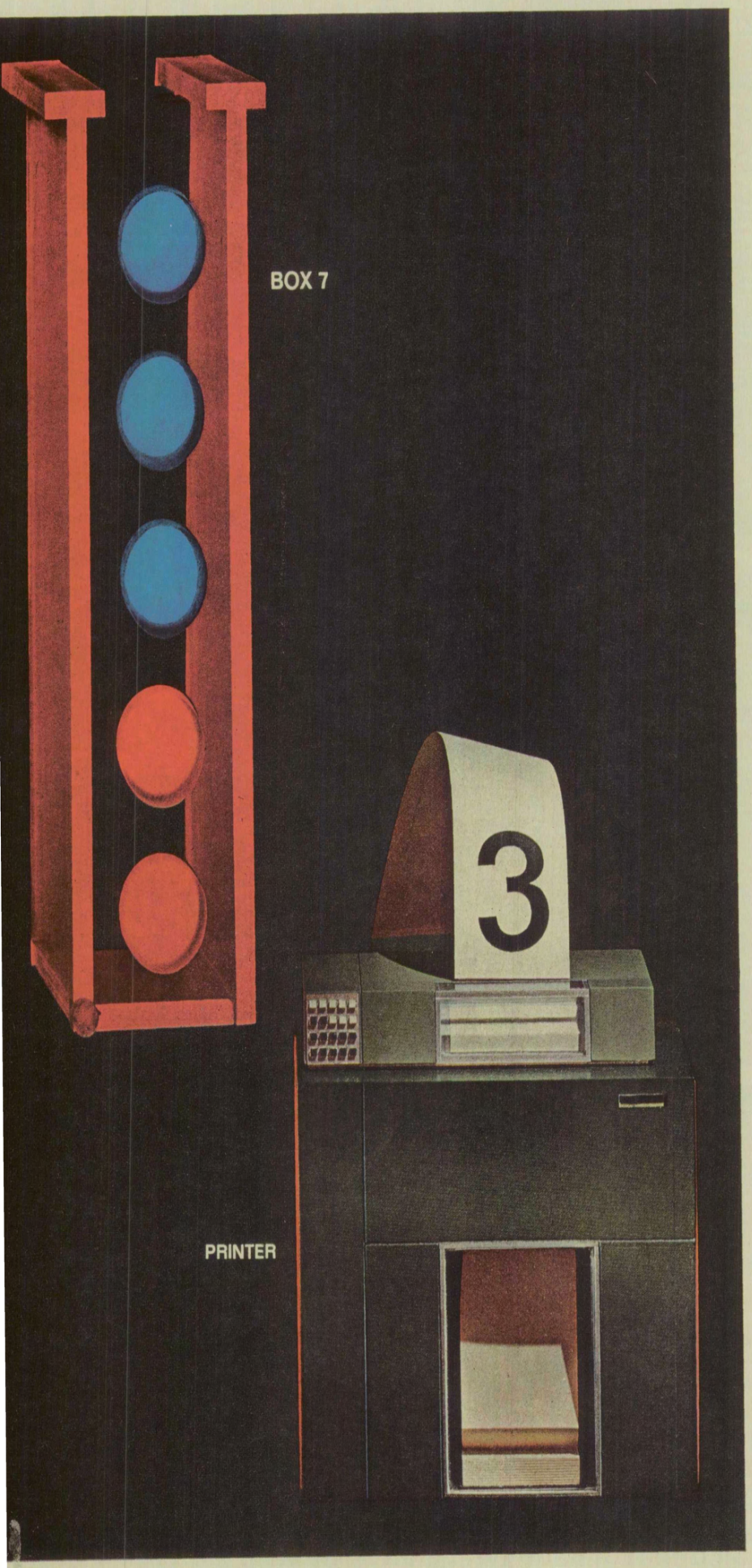
In the Accumulator of our railroad computer, a result has been reached by this method and is held in temporary storage.

Having completed two trips, the train is now back at Box 3, where it receives the order: "Take the number now in the Accumulator and store it in Box 7." The two-disk signal for "store" is put down at the Operations Codes turntable and the Box Address is deposited at the Box Addresses turntable. The train then proceeds via Track 7 all the way to the Operations Codes turntable, which directs it out along the Store track to the Accumulator. Here it receives what the Accumulator has accumulated, namely two red disks followed by three blue ones. The train continues down the track to the crane beside the Answer box. Here the answer is unloaded and stored in Box 7. From there the train circles back for its final instruction, provided by Box 4. This instruction reads: "Take what is in Box 7 and carry it to the Printer." Following a now familiar pattern, the train goes to Box 7, picks up its contents and moves to the Operations turntable and out along the Print track to the Printer where another crane waits to unload its cargo.



A model train tracks the route to a solution

The real thing: tapes, cores, logic units



Once our train has carried the answer to the Printer, its job is done. It remains only for the Printer to convert the final binary number into a form that the operator can read. Using a device like the Encoder (*pages 62, 63*), it swiftly does just that—and prints out the number 3 (*left*).

How the railroad's boxes, turntables, track loops and other equipment look in a real computer is shown at right. This is one of the newest and most versatile machines, Model 40 in IBM's System/360 line. It is opened up to reveal its electronic components. It would have performed each of the steps in our problem in five millionths of a second.

At the upper right is the control panel, which enables the operator to alter the program, interrupt the computer if need be or request progress reports on his problem. The typewriter console at top left can print out such reports and also provide additional access to the computer. Actually, Model 40 can take on the problems of 63 separate operators, provided each has his own device for communicating with the machine. These devices can be in the next county, the next state—or another country, for that matter.

Unlike our model railroad, which had a single group of boxes in its Memory section, Model 40 has two areas where massive amounts of information can be stored. One chassis, shown at lower right, holds permanent instructions. A second, appearing above and to the left, contains data and instructions for any specific program being run.

The permanent instructions handle operational procedures likely to come up no matter what the problem: addition, subtraction, finding the square root of a number, and so on. When one of these instructions is encountered in a program, the machine switches automatically to it to determine how to proceed. These instructions are encoded on wire-laced plastic tapes laid one on top of the other in a cartridge (bottom right). The wiring runs down each tape in vertical columns, and locations along the network are either "on" or "off." When one of these tapes is called on, the "ons" and "offs" are read in sequence, and an instruction is produced.

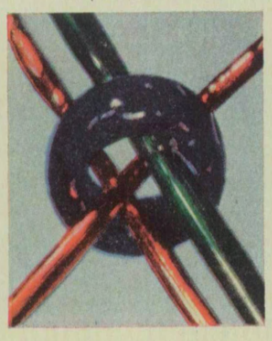
For the storage of data and program instructions, a different system is used, comparable to the memory boxes of our model train. Its basic device is the magnetic core, a ring of ferrite 3/100ths of an inch in diameter (*right*). Cores can be magnetized in either a clockwise or a counterclockwise direction—again the equivalent of the "on-off" or 1 - 0 binary notation. The cores are strung on frames by pairs of wires that intersect each core at right angles, giving the completed frame (or core plane) the appearance of an ambitious exercise in beadstringing. Each of the 33,280 cores in a plane has its own "address," determined by the two wires that intersect it like coordinates on a map. Nine core planes are stacked one on top of the other to form a core array.

There is in the first core plane, therefore, one core that is identified by a particular set of coordinates. There is another core at the same two coordinates in the second plane and one in each plane through the ninth. A reading of the stack's nine similarly identified cores forms a nine-bit unit of information.

When information is being put into the stack, the direction of the current flow in any pair of intersecting wires sets which way that particular core is magnetized. A third strand, called a sensing wire, is woven through every core in a plane. These wires report on the magnetic state of cores when information is being taken out of the stack. The whole operation is incredibly fast: the sensing wires announce the "off-on" state of cores at the rate of 6,400 every thousandth of a second.

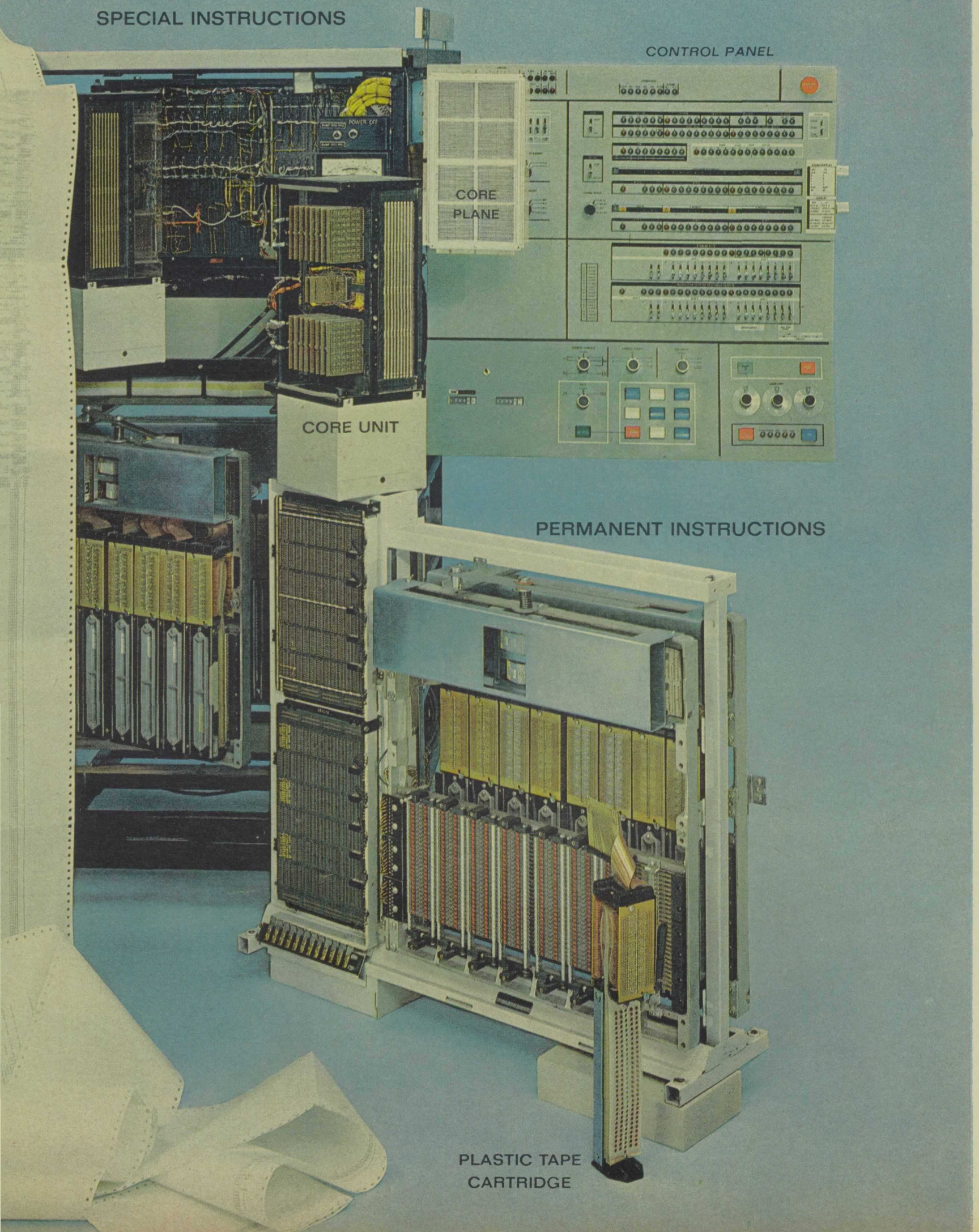
At the lower left is the computer's Arithmetic unit, consisting of thousands of transistors, resistors and diodes which can perform additions at a rate of 100,000 a second. Because the Arithmetic unit can also be programmed to make certain choices and comparisons —comparing the size of two numbers and selecting the next step on that basis—it is also known as the logic unit. Behind it is the Control section of the machine—the track and turntables of our model—with panels of wiring and switching apparatus that connect all the units of Model 40 together.

CORE (MAGNIFIED 50 TIMES)



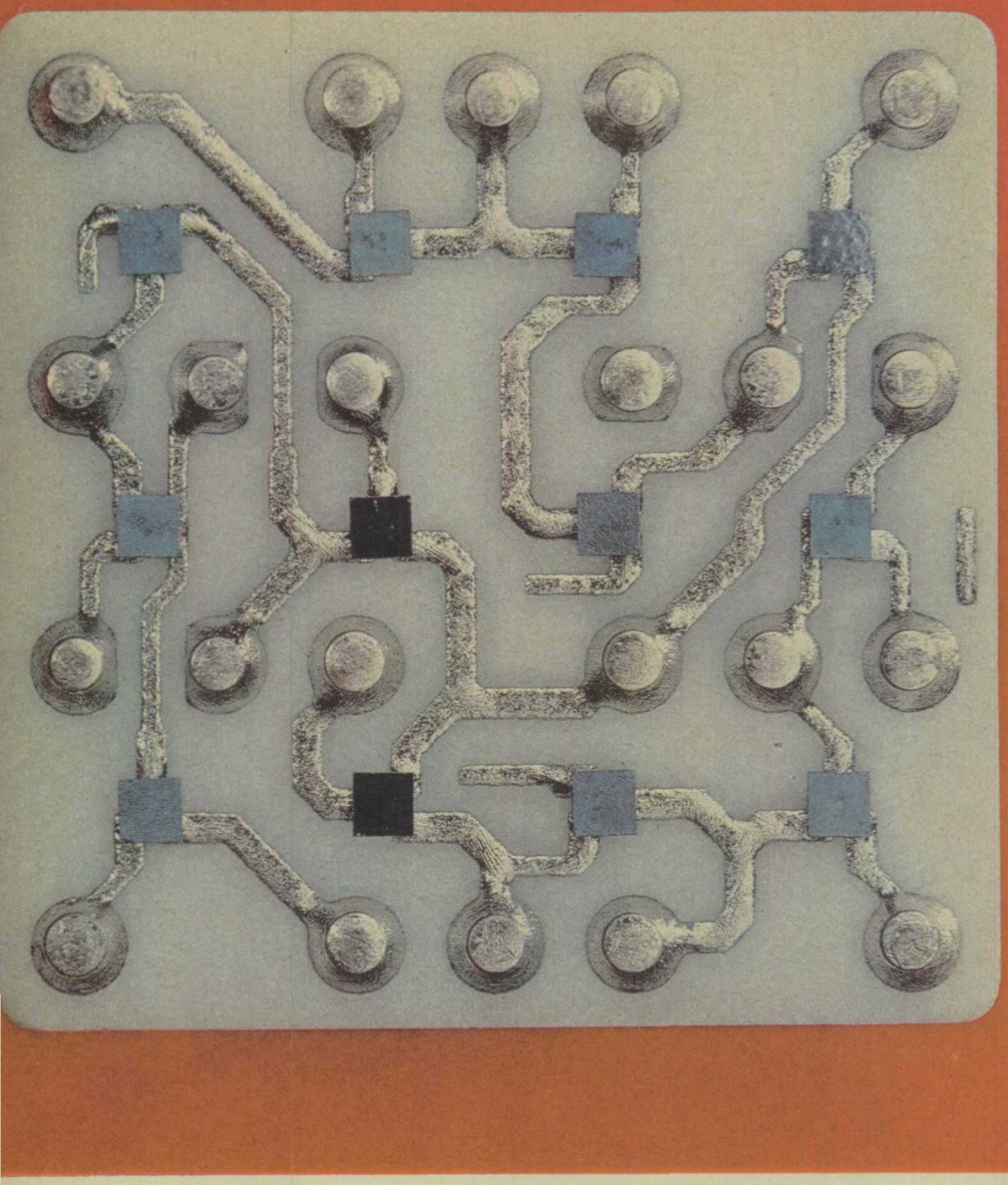
MEMORY

DATA PLUS SPECIAL INSTRUCTIONS

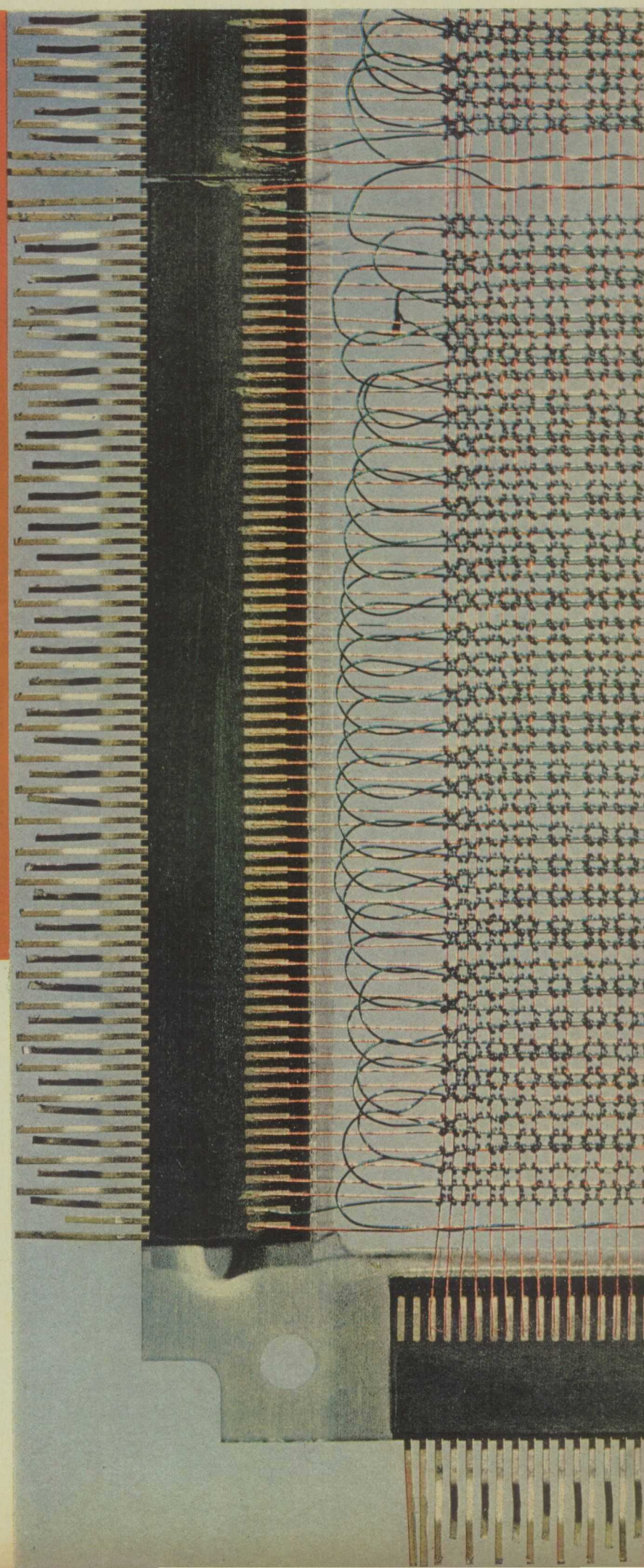


Components invariably betray a computer's age, and the ones shown here are brand-new. The first machines used great banks of vacuum tubes—the kind found in radios—which took up a large amount of room, consumed enormous amounts of power and were not, by modern standards, very efficient at all. In the

A new generation of parts to



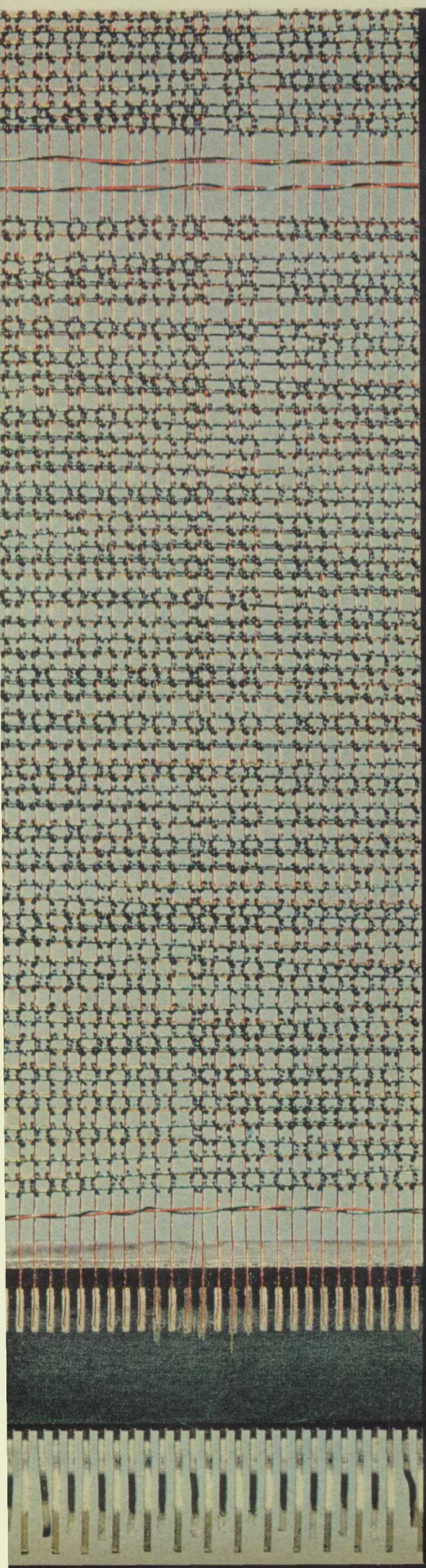
Miniature circuit, shown enlarged 10 times, is a basic element of the System/360. Lines of conductive metal printed on a ceramic base replace wiring and connect the elements on it. The squares are transistors and diodes, .028 of an inch across, which act as electronic switches. The blobs are the soldered tops of conductive connecting pins which protrude from the bottom of the unit and are used to plug it into place in the computer.

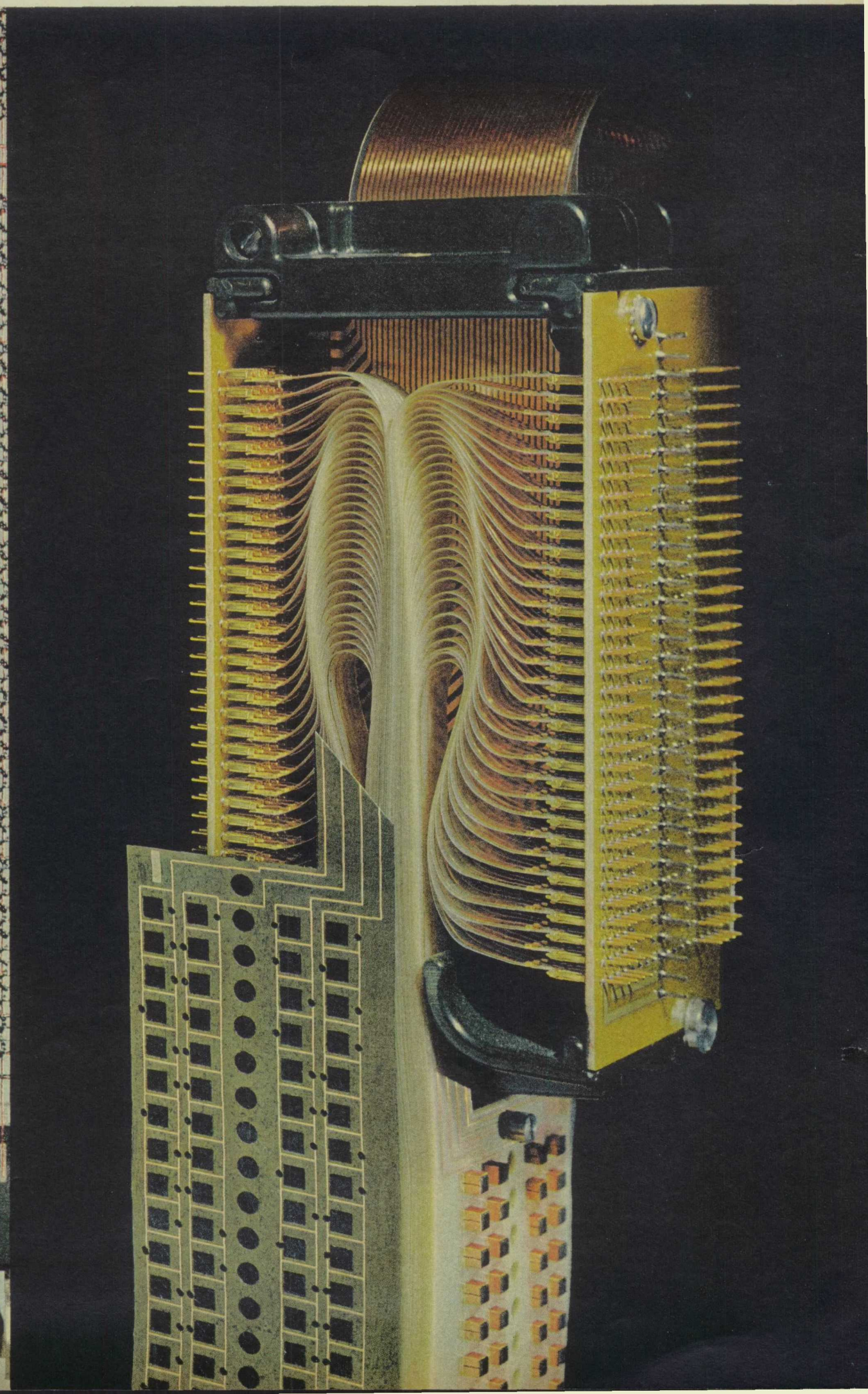


An enlarged portion of a core plane (*right*) from System/360 shows the tiny ferrite cores strung onto wires that intersect at right angles. Laced through the grid is the sensing wire that reads the cores. At the far right is a close-up of a tape cartridge which holds the computer's permanent instructions. In front of the unit is a single tape, showing conductive copper lines that branch as they lead downward, each branch passing a column of square holes either on the inside or the outside. If the inside branch is unbroken at a hole, the location reads "on." If the line is cut, the message is "off."

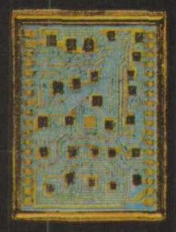
second generation, tubes were replaced by transistors, which were so much better in every respect that computer designers began to think about a new problem—speed. For a pulse to travel over eight inches of wire takes one nanosecond—a mere one-billionth of a second, but an interval large enough to create difficulties. More compact circuits, like the one at the left, with transistors cut down in size, shaved the distance that pulses had to travel. This process of shrinking circuitry, called microminiaturization, marks the computer's entry into a third generation—and units on the next page represent a step further still.

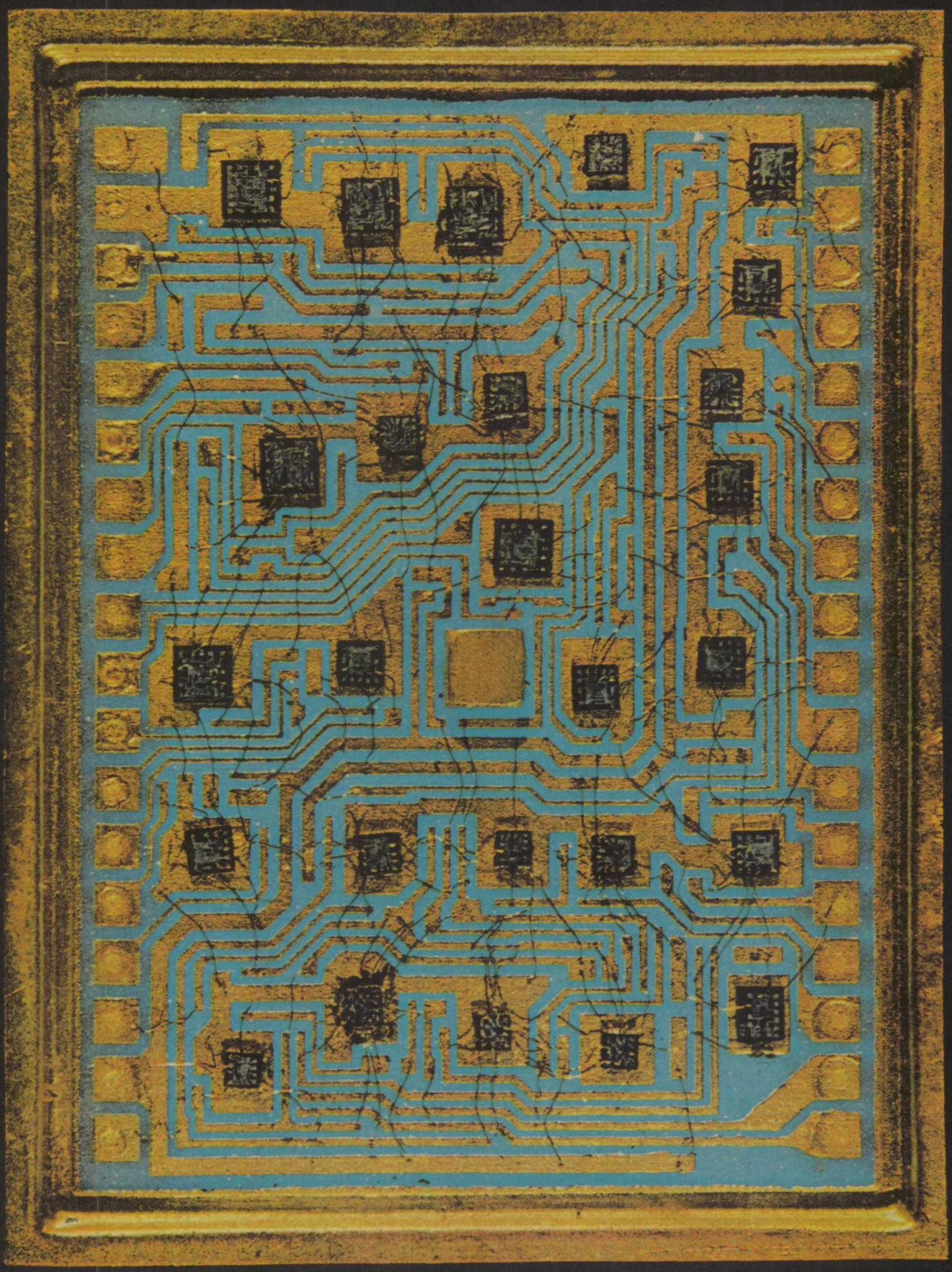
save billionths of a second

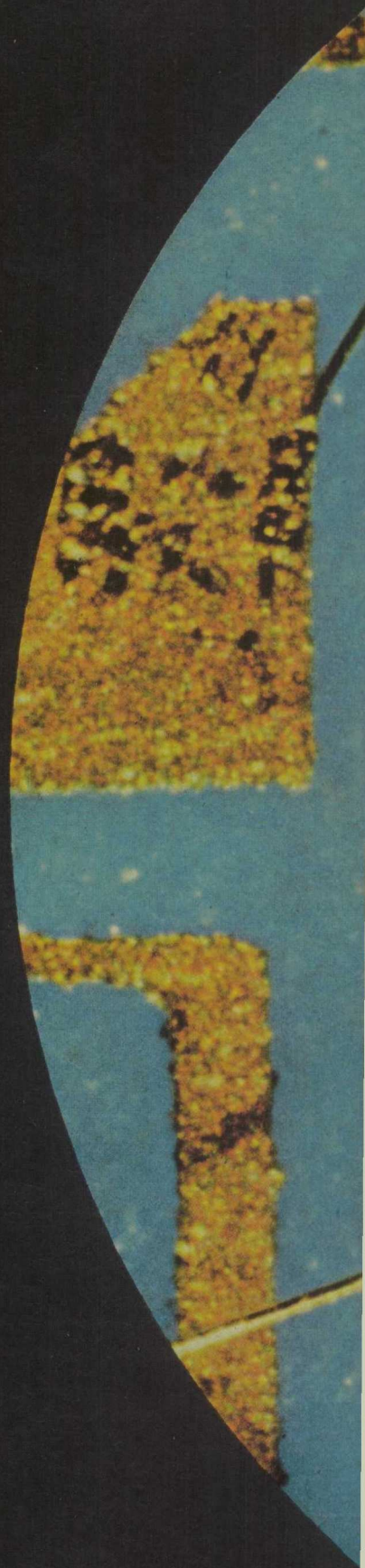


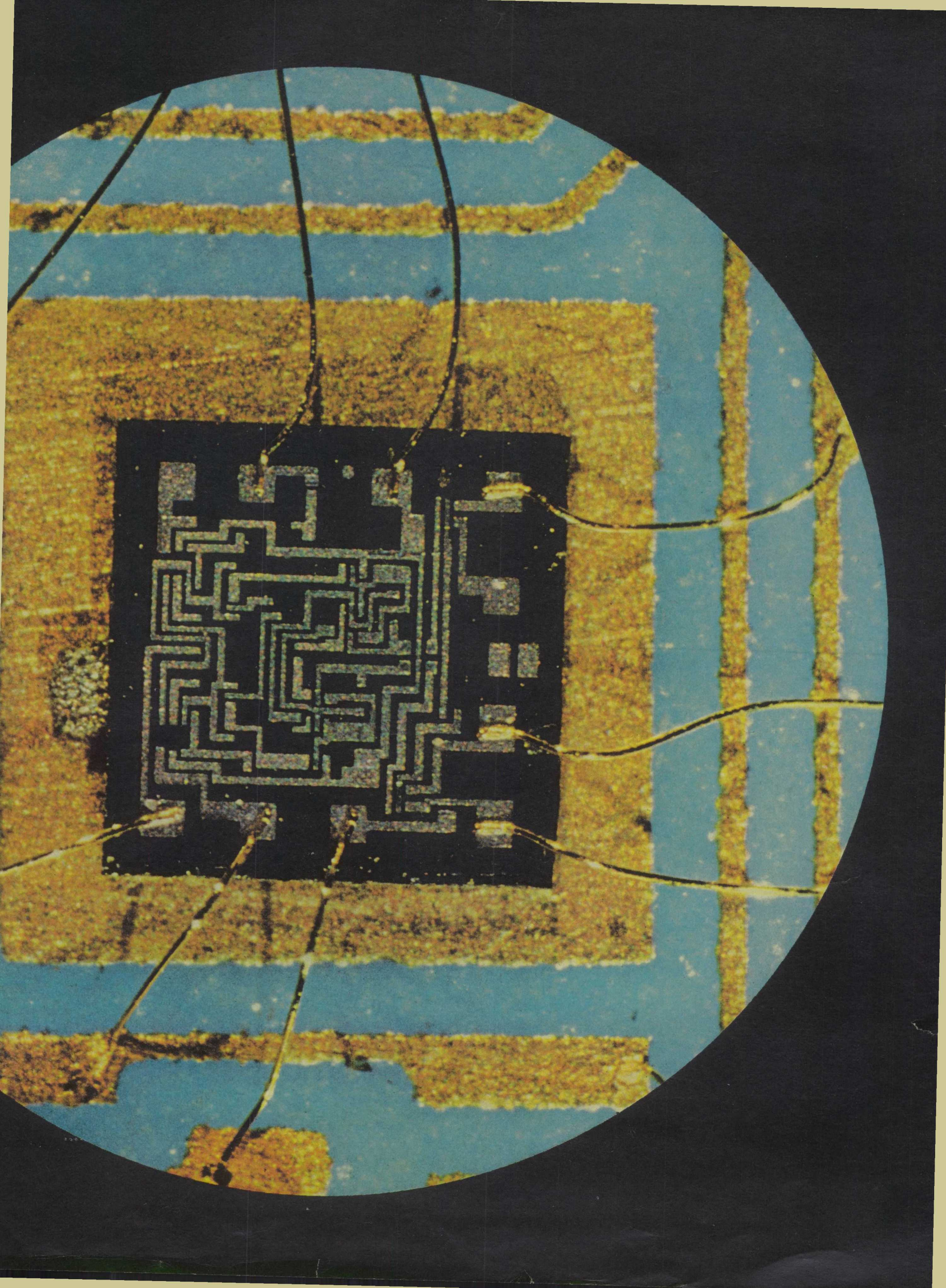


With the discovery that shrinking computers also increases their efficiency and speed, microminiaturization has become one of the fastest-growing fields in the industry. The size of some special-purpose machines has already been reduced from that of a refrigerator to something more like a suitcase. The switching unit at the left (shown actual size there and much enlarged below) is actually several times larger than the ceramic plaque seen on the previous page. Yet each of the 26 tiny chips on its surface contains as much circuitry as the whole of that plaque. A single such chip (seen through a microscope at the right) is the equal of all the circuits in an eight-transistor radio. And in the whole assembly there are enough circuits for three television sets—tubes, wiring, sound and all. Besides driving down costs and increasing reliability, miniaturization insures the computer's proliferation toward a time —not so far off—when even the home will utilize the machines.









The eerie interface of man and machine

Can a computing machine be taught to "think" in the broadest sense, as the human brain can think and learn? Efforts have been made in this direction, based on accumulating knowledge of how the human brain works. The brain is, in effect, a whole hierarchy of computers with much of its basic input provided by ranks of lesser computers down the line. The information gathered by the retina of the eye, for instance, is not simply shunted along to the brain. Instead, the retina acts like a tiny, highly sophisticated computer, analyzing the visual data and passing on only the significant results. In the brain itself further analysis takes place as information moves from one level to another and is processed in computerlike fashion all along the way until a unified perception results.

In a computer, electric pulses travel to specific places along specific conduits to produce their own version of a unified perception in the form of an answer. Not so in the brain. One brain cell, or neuron, may pulse another, as the electronic switches in a computer pulse each other, but the similarity ends there. Each neuron has inputs from a number of other nerve cells, so that in one way or another all the 10 billion or so neurons in the brain are interconnected directly or indirectly. The system provides whole arrays of incredibly complex feedback loops in which some cells qualify the operations of others, stimulating them to respond to certain incoming signals and inhibiting their responses to others.

Theoretically it would be possible, as one computer scientist has noted, to hook together hierarchies of computers to simulate the complex layers of the brain. The gear necessary to give this supercomputer even 1/20th of a human brain's capacity would fill several barns. And the fact is that no one would really know how to hook together such an array. Scientists have a general idea about how the brain functions, but a wiring diagram, or anything like it, is almost totally lacking and seems likely to remain elusive for a long time to come. Even if a wiring diagram came to hand, the problem of writing a program for such a contraption would present an equally enormous obstacle. Reflecting on the difficulties, a veteran programmer said, "It takes us 15 to 20 years to program our children. And they can *really* learn —sometimes. But with the kind of machine you're talking about, you would have to feed in millions of separate little things every day to equal what a child takes in. And after three years of this kind of business, it still might not have learned to reason or work out its own programs. You ask it to add 2 and 2, and maybe it says 5."

Since "thinking" is apparently not a prospect—at least not until the brain is better understood or a major new breakthrough revolutionizes computer technology again—computers will remain painfully literal-minded. One important word in the lexicon of computer programmers is GIGO, which simply means: "Garbage In, Garbage Out"—that is, if the wrong information is put into the machine, the results will also be wrong. The computer has no sense that it is being fed garbage, and anyone who gets about much on the computer circuit is bound to hear some rich examples of GIGOism.

At Cape Kennedy some time ago a rocket rose majestically from its launch pad and headed toward Venus. But following liftoff it began to wobble uncontrollably and had to be destroyed after only 293 seconds of flight. What had happened to $18.5 million worth of hardware was that a programmer omitted a minor item from the long list of explicit instructions to the computer that was "flying" the bird.

The missed item was a hyphen, and its omission led the computer into the same sort of misunderstanding that leaving the hyphen out of a phrase involving "fifty-five gallon drums" might have on a human.

When Astronauts L. Gordon Cooper and Charles Conrad splashed down 103 miles off target on Aug. 29, 1965, it was no fault of theirs or of their computer. The flight, including reentry, was computer-guided. In determining the exact time for firing retro rockets, the programmer had assumed that the earth revolved exactly once every 24 hours, whereas in fact—as we know from having to squeeze in a whole extra day every fourth year—it makes a shade more than one revolution in that time. But if you're orbiting the earth many times and someone fires the retros exactly at 1:51 p.m., after figuring on a day of precisely 24 hours, you can wind up out of touch with the earth by a significant number of earthy miles—which is just what happened to Astronauts Cooper and Conrad.

The very simple-mindedness of the computer plays a significant part in its highly symbiotic relationship with humans. It is everything we are not. Human beings, by and large, are imaginative, intuitive, sensitive to values and occasionally capricious. A computer is none of these things. It can pay undivided attention to details that would drive a mortal right out of his mind. It can be told something and won't forget it until told to do so. It never gets huffy. It will work on the most boring problem forever without getting overheated. It will not laugh outright at human error and will work prodigiously at any problem put to it, no matter how trivial.

People who work with computers—especially those being exposed to a machine for the first time—can become quite entranced with these qualities, finding in the computer a kind of alter ego. "Sometimes programmers just won't go home, take a bath or anything," reports a computer man who has got over it himself. "They're like a kid falling in love with a hot rod. They'll sit there working with their newfound 'friend' 20 hours a day, just watching the lights and drinking coffee. After a while they get to looking pale and unhealthy. They sit there fascinated and just forget to eat."

This affectionate relationship between man and machine takes its most acute form in major computer centers, where many people learning programming and using computers come together to work with one or a cluster of the big machines. Inevitably, the idea is to think up something new for the computer to do. Here both students and instructors devise rather fanciful programs that pit the machine's quick, literal and retentive mind against the inventiveness of humans. The results of such interaction can be quite impressive.

In one such center a game was devised with the program title of HR-3. When the operator typed in "HR-3," five rows of tiny lights, extending across the display console of the computer, were activated. In any given row a light would go on, then out, and the light to its right would be illuminated—and so on in all five rows from one side of the panel to the other. How fast the flashing lights moved across any row was determined by a program called a "random number generator," which has many uses in higher mathematics but was here employed to bump the lights along in random fashion. The group would bet on which row of lights would reach to the end of the console first. The winner bought morning coffee. What they had, with the aid of several million dollars' worth of computer, was a completely honest race, which accounts for the title of the program, "Horse Race 3."

A more complex program, which many centers worked out, involved teaching a computer to play blackjack. Anyone with access to the machine could type in the code word for the program, and the computer would type back:

WELCOME TO THE GAME OF BLACKJACK.
I AM GOING TO DEAL.
WHAT IS YOUR BET?

As the game progressed, the computer also kept a running tally of wins and losses, and would report this to any individual player even if he dialed a month later. A computer never forgets.

Such gambits are generally ill-regarded by the managers and bosses of computer centers—even if the purpose is training, computer time is still extremely expensive. It is said that a general in charge of one computer discovered the continuing blackjack game and ordered it erased from the computer's memory. This was obediently done. But the general didn't think to have the original program classified and burned, so it was surreptitiously put back in the machine—under a new heading in the program index. It may be there today, half forgotten among programs of great national import. Which brings up the fascinating (and quite real) possibility that someday an operator who wants the latest computerized estimate of China's nuclear capability—access Code BJ-20—will type into the machine instead BJ-21. And from the unforgetful machine will come the reply:

WELCOME TO THE GAME OF BLACKJACK.
I AM GOING TO DEAL....

TYPEWRITER CONSOLE
CONTROL
ARITHMETIC
(LOGIC)

© Life Inc., 1967. All rights reserved. No part of this material may be duplicated or redisseminated without permission.